Tip

The `Option Explicit` statement forces all variables to be declared before they are used. If `Option Explicit` is omitted, variables are automatically defined at first use as type `Variant`.

Listing 2. Function that returns the value 5

```
REM  *****  BASIC  *****
Option Explicit

Sub Main

End Sub

Function NumberFive ()
  NumberFive = 5
End Function
```

8) Within the LibreOffice Basic IDE select **File > Save** on the Menu bar, or click the **Save** icon on the Standard toolbar, or press *Ctrl+C*, to save the modified Module1.

Using a macro as a function

Using your newly created `CalcTestMacros.ods` spreadsheet, select a cell and enter the formula `=NumberFive()` (Figure 437). Calc finds the macro, calls it, and displays the result (5) in that cell.

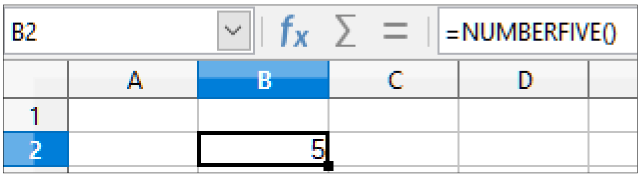

Figure 437: Use the NumberFive macro as a Calc function

Tip

Function names are not case sensitive. In Figure 437, the function name was entered as `NumberFive()` but Calc displays it as `NUMBERFIVE()` in the Formula bar.

Macro security warnings

You should now save the Calc document, close it, and open it again. Depending on your settings in the Macro Security dialog accessed using **Tools > Options > LibreOffice > Security > Macro Security** on the Menu bar, Calc may display one of the warnings shown in Figures 438 and 439.

In the case of the warning shown in Figure 438, you will need to click **Enable Macros**, or Calc will not allow any macros to be run in the document. If you do not expect a document to contain a macro, it is safer to click **Disable Macros** in case the macro is a virus.

In the case of the warning shown in Figure 439, Calc will not allow any macros to be run in the document and you should click the **OK** button to remove the warning from the screen.

When the document loads with macros disabled, Calc will not be able to find any macro functions and will indicate an error in any affected cell by displaying the text `#NAME?` in that cell.